



Design and Implementation of an Intranet Operations Board

BSc Computing (Networks & Communications)
Ben Mason
2003



Sheffield Hallam University

Acknowledgements

Many thanks to Chris Bates for his support.

Thanks to Mervyn Sloan for helping me put my idea forward.

Thanks to the Operations Team in IT Operations, Sun Microsystems GmbH, for their support, feedback and help throughout the implementation of the Operations Board.

Thanks also to my father for proof reading my Dissertation.

Abstract

This project report details the design, development and implementation of an Intranet Operations Board for Sun Microsystems GmbH. An electronic Operations Board was designed to replace an existing task assignment system using a physical whiteboard. Incoming maintenance and service tasks are assigned by a Duty Manager dependant upon their priority, severity and type. The need to access and view the data from sites all over Europe and Africa necessitated an electronic solution to what had previously been a physical in-house system. Various software and development solutions were considered, finally arriving at a viable solution which could be implemented efficiently with minimum disruption to the day-to-day working of the IT Operations department. The Operations Board was brought on-line side-by-side with the existing system until a fully usable solution was produced. Ultimately the old system was discarded and the new system became the standard. This report describes the process by which the system, software and methodologies of the new Operations Board were derived.

DYNAMIC WEB BASED INTRANET DATABASE INTERFACE

Table of Contents

1.	Introduction.....	1
2.	Investigation & Analysis.....	2
2.1.	Identification of the problem context.....	2
2.2.	Background investigation and references.....	3
2.2.1.	Web usability.....	3
2.2.2.	Dynamic web sites.....	4
2.3.	Investigation of appropriate tools and methodologies.....	5
2.3.1.	Web development technologies.....	5
2.3.1.1.	Perl.....	5
2.3.1.2.	Macromedia ColdFusion.....	6
2.3.1.3.	The Apache HTTP Web Server with PHP.....	7
2.3.1.4.	Z Object Publishing Environment (Zope).....	8
2.3.1.5.	Selection of web development technology.....	10
2.3.2.	Database technology.....	11
2.3.2.1.	Oracle.....	11
2.3.2.2.	MySQL.....	11
2.3.2.3.	Gadfly.....	12
2.3.2.4.	SAPDB.....	12
2.3.2.5.	Selection of database technology.....	13
2.4.	Investigation of process models.....	13
2.4.1.	Waterfall model.....	14
2.4.2.	Spiral model.....	15
2.4.3.	Rapid prototyping.....	16
2.4.4.	Selection and justification of model.....	17
2.5.	Investigation of design methodologies.....	17
2.5.1.	Structured design methodology.....	17
2.5.2.	Object-oriented methodology.....	18
2.5.3.	User Centred Design (UCD) methodology.....	18
2.5.4.	Top-down methodology.....	19
2.5.5.	Bottom-up methodology.....	20

2.5.6. Selection and justification of methodology	20
3. Design of the Operations Board	21
3.1. Database/table creation	21
3.1.1. The 'user' table	21
3.1.2. The 'tasks' table	22
3.1.3. The 'hosts' table	24
3.2. Interface/GUI creation	26
3.2.1. Security	26
3.2.2. Look and Feel	27
3.2.3. Code modules.....	31
3.2.3.1. The Main page	34
3.2.3.2. Adding a Task	35
3.2.3.3. Viewing a Task.....	36
3.2.3.4. Updating a Task.....	37
3.2.3.5. Searching for a Task.....	38
3.2.4. The Whiteboard	40
3.2.5. Constraints.....	42
4. Development.....	44
4.1. Implementation of the Operations Board	44
4.2. Validation and verification (V&V)	46
4.2.1. Validation - Are we building the correct product?	46
4.2.2. Verification - Are we building the product right?	47
4.3. User feedback	47
5. Reflections.....	49
5.1. Completeness of the Operations Board.....	49
5.2. Strengths and weaknesses of the Operations Board	49
5.2.1. Strengths	50
5.2.2. Weaknesses.....	51
5.3. Further tasks to enhance the Operations Board	52
6. Critical Evaluation	53
7. References	56

8. Bibliography	57
9. Appendix	58

1. Introduction

This dissertation details the design, implementation and testing of an Intranet Operations Board. The idea of the Operations Board was produced during the developer's placement year with Sun Microsystems. The subject of a web based Operations Board was chosen because it is of particular interest to the developer.

Web usability, web technologies and database technologies will be discussed when deciding on the correct approach to be taken. Other aspects such as design methodologies and process models will also be analysed. The methodology will be chosen and justified, as will the process model. This methodology will be then put into practice by designing and implementing the Operations Board.

The strengths and weaknesses of the final deliverable will be discussed, as well as future enhancements of the tool.

The *Project Specification* can be found in Appendix A1.

2. Investigation & Analysis

2.1. Identification of the problem context

The author's placement year employment was in an internal IT support group for Sun Microsystems GmbH, known as World Wide Client Services IT Operations, and referred to after this point as IT Operations. Part of the work involved delegating and organising maintenance tasks, as well as dealing with Outages. An outage is the duration of time a system or service is unavailable. There are two types of outage, planned and unplanned. A planned outage is usually created by scheduled maintenance. An unplanned outage is when a system or service becomes unavailable due to a hardware, software or environmental fault. Scheduled maintenance may include Operating System upgrades, server hardware repairs or new service installations.

At present, the department employs a Duty Manager to delegate the maintenance tasks and organise the outages. The computer systems are configured to E-mail the Duty Manager whenever a fault occurs. This means that all of the information is dealt with centrally. Tasks of greater importance are dealt with by the Duty Manager. This may involve creating a Global Change Notification (GCN). A GCN is a means of informing users of a current or future outage affecting them. The GCN includes the start and end time as well as detailed information about what is being done to resolve the problem. Tasks which have been deemed less important by the Operations Manager are sent directly to the current Operations Team. The Operations Team consists of up to five people who deal with the daily maintenance tasks. The Duty Manager is expected to delegate the majority of his work, as he needs to be available to deal with high priority issues should they arise.

IT Operations supports internal users throughout Germany, Austria and Switzerland. Approximately 2000 users are supported by a team of fifteen people, including the five operations team members. Presently, outages and on-going maintenance are displayed on a whiteboard using whiteboard marker pens. This method was fine whilst the Operations Team was completely based in the same location. After a new recruit was to be placed in the Berlin office, and was to occasionally be a member of the operations team, the need for a new method of displaying, and communicating, the operations team's tasks was apparent.

The developer, being a member of the Operations Team for the majority of his placement, decided to design and implement a new system. The concept of this project was to design and implement a central intranet Operations Board. After discussion with the Technical and Region Managers, this concept was accepted. It was decided that the Operations Board would initially run along side the existing system, but then replace it. The Operations Board would be a dynamic web based front-end for a database.

2.2. Background investigation and references

2.2.1. Web usability

The usefulness of a tool such as the Operations Board will depend highly on usability. If a new system is overly complicated or difficult to use then users will be reluctant to adopt it. There are several key issues when designing a web site or portal. The issues below have been taken from The Web Style Guide (<http://www.webstyleguide.com>).

Navigation

Most user interaction with a web site involves navigating hypertext links. Users should always be able to navigate their way to the homepage. When users bookmark a particular page in the web site, they will return

to that one page. If they then cannot get to the main page easily, then these pages are just 'dead-ends' that prevent the user exploring the web site correctly.

Direct Access

Users want to get information in the fewest steps possible. This means that an efficient hierarchy of information must be designed. Studies have shown that users would prefer a few dense screens of choice, as opposed to many simple choices.

Bandwidth

Web sites should be designed with the bandwidth of the users connection in mind. For example, if a user is using a 56K dial-up then there is little point putting large images on the web pages. If the site is being designed where the users will be accessing it at ethernet speeds, then it can be more ambitious with its content.

Simplicity and consistency

Users do not like overly complex web sites. They become used to the big corporations' style of web site, with their easy to navigate and less clutter approaches. Each web page should have the same look and feel as the previous one.

2.2.2. Dynamic web sites

Dynamic customised web content is becoming very widespread, from web based E-mail to corporate web sites. In order for sites to store and recall a user's preferences, they must use a database system. Any site, therefore, that is storing and recalling user's details, is using a web driven database system.

An example of a site using such a system is the Sheffield Hallam University Learning Centre. The user, after providing their patron number and PIN, is presented with their library records. The user is able to see what books they currently have on loan, which ones they have reserved and are able to renew current loans. This library system uses a web server to provide dynamic content based on the contents of a database.

2.3. Investigation of appropriate tools and methodologies

There are two parts to the Operations Board; the web interface and the database system. In order for the Operations Board to operate efficiently, these systems must work **intimately** with each other. The different web development and database technologies are investigated and detailed in the new few sections.

2.3.1. Web development technologies

There are many web development tools available to use. As the Operation Board is being created for Sun Microsystems, it must run on UNIX. Another aspect to be considered is that **no** Microsoft software can be used as Microsoft is one of Sun's direct competitors. This then prevents the use of software such as Microsoft Access for a database system, and Microsoft IIS as a web server technology. With these aspects in mind, relevant web development technologies are detailed below.

2.3.1.1. Perl

Perl is a script programming language that is similar in syntax to the C programming language. Perl is an interpreted language, which means it does not need to be compiled prior to execution.

Perl is regarded as a good choice for developing common gateway interface (CGI) programs because it has good text manipulation facilities, although it also handles binary files (WhatIs?Com, 2003).

In order for the developer to access a database within Perl, a Database Interface (DBI) must be used. DBI is designed to obviate the need for knowledge of vendor libraries. It has a very simple interface for taking Structured Query Language (SQL), and returning results. DBI doesn't know how to talk to any particular database, but it does know how to locate and load in Database Driver (DBD) modules. The DBD modules have the vendor libraries in them and they know how to talk to the real databases; there is one DBD module for every different database (perl.com, a short guide to DBI, 1999).

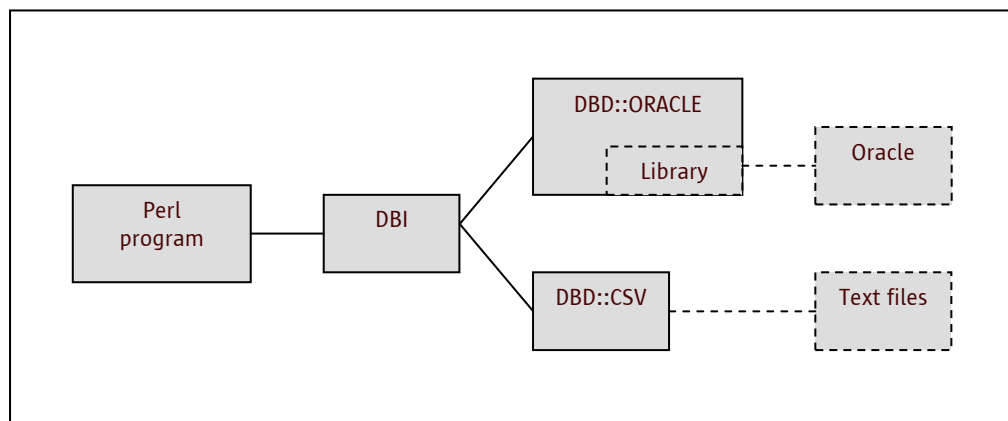


Figure 1 - How a Perl program connects to an Oracle database or a comma separated value (CSV) text file using DBI and its driver database (DBD).

2.3.1.2. Macromedia ColdFusion

ColdFusion, developed by Allaire which has recently merged with Macromedia, is a popular and sophisticated set of products for building Web sites and serving pages to users. With ColdFusion, a company can build a content database using input templates and combine these with application programs to create a Web site in which pages are developed dynamically as

they are served. The ColdFusion package consists of ColdFusion Studio, which is used to build a site, and ColdFusion Server, which serves the pages to users. ColdFusion Studio is described as "a complete integrated development environment (IDE)" and ColdFusion Server as "a deployment platform" (searchDatabase.com definition, 2003).

ColdFusion allows development of high-end, scalable web applications using the ColdFusion Mark-up Language (CFML), which is ideal for programming web applications. Processed entirely on the server, CFML tags cleanly integrate with HTML for user interface and XML for data exchange. Both open and extendible, CFML supports more than 70 server-side tags, 200 functions, and 800 third-party components (iWebPro.com, 2002).

Components can be created easily in ColdFusion, using its built-in Component Creator. These components can be rapidly deployed and re-used.

ColdFusion works on both Windows and UNIX. The Windows version allows it to connect to a Microsoft Access and Microsoft SQL, whereas the UNIX version will connect with MySQL.

2.3.1.3. The Apache HTTP Web Server with PHP

Apache is a powerful and flexible web server which implements the latest protocols. It is highly configurable and extensible with third-party modules. Modules can be written using the Apache module API. Apache is Open Source, which means its source code is freely available for use or modification as users or developers see fit. It runs on a multitude of different operating systems including UNIX. Authentication modules allow password-protected pages with many authorized users, so security is not a problem, and is easily configurable (Apache, 2002).

Apache is the fastest web server available for serving static web pages. It is also the most popular, according to the Netcraft Web Server Survey of February 2003, Apache provided over 62% of the internet's content.

Apache alone cannot deliver database content to the web; a third-party module must be used. One of the available modules is the Hypertext Pre-Processor, known as PHP.

PHP offers a simple and universal solution for easy-to-program dynamic web pages. The intuitive interface allows programmers to embed PHP commands right in the HTML page. PHP's syntax is similar to that of C and Perl, making it easy to learn for anyone with basic programming skills. Its elegant design makes PHP significantly easier to maintain and update than comparable scripts in other languages. PHP offers excellent connectivity to most of the common databases (including Oracle, Sybase, MySQL, ODBC and many others), and offers integration with various external libraries, which allow the developer to do anything from generating PDF documents to parsing XML (Zend - The PHP Company, 2002).

2.3.1.4. Z Object Publishing Environment (Zope)

Z Object Publishing Environment (Zope) is an open source Web application platform from Digital Creations (<http://www.zope.org>). Zope is freely available for UNIX, Linux and Windows and will interoperate with Web servers such as Apache and IIS. It competes with middleware products such as Cold Fusion, Netscape Application Server, PHP, mod perl, Frontier and Vignette (Zope, 2002). Some of the features of Zope are detailed below:

Unique Management Environment

Zope allows the sites data, logic, and presentation to be managed from a web browser. Zope is easy to use and is remotely administrable. Zope allows many users to collaborate to interactively develop the web site.

Web server

Zope comes with a built in web server that serves content to the users. Zope integrates easily with an existing web server, such as Apache or Microsoft IIS.

Web based interface

Applications built in Zope use the web browser to interact with the Zope management interface, which provides the developer with many tools.

Relational integration

In order to interface with external data sources such as an Oracle or MySQL database, Zope must use adaptors. Adaptors are third-party modules that give Zope more functionality. The MySQLDA is an adaptor that provides MySQL support for Zope.

Scripting language support

Zope contains support for different scripting languages, such as Python, Perl and the Document Template Markup Language (DTML).

2.3.1.5. Selection of web development technology

It is clear to see from the above details that all of the web development technologies contain many features. The features can be extended with the use of third-party modules. The best way compare the technologies, is to use a matrix.

	Perl	ColdFusion	Apache & PHP	Zope
Works on UNIX	YES	YES	YES	YES
Open source	YES		YES	YES
Database integration	YES	YES	YES	YES
Built-in web server		YES	YES	YES
Dynamic page templates	YES	YES	YES	YES
Scripting support	YES	YES	YES	YES
Through the web editing		YES		YES
Object oriented		YES		YES

Chosen web development technology: Zope

Zope was chosen because of five main features:

Firstly, its ‘through-the-web’ editing, allowing the site to be modified from anywhere, using a standard web browser; secondly, its object oriented approach, which allows documents or data to be moved around, copied and deleted easily; thirdly the built-in database connection allows integration with a database system, such as Oracle or MySQL; fourthly, the dynamic HTML template language allows dynamic web content to be created from sources such as databases and scripts; fifthly, Zope is free to download and does not require a license to run.

Zope will be using its built-in web server to serve the pages, being the most reliable and the quickest way.

2.3.2. Database technology

The two main aspects to consider when investigating a database technology is its ease of use, and its compatibility with the chosen web development technology. Some of these technologies are described below.

2.3.2.1. Oracle

Oracle is a well established database system. Oracle provides efficient, reliable, secure data management for high-end applications such as high volume on-line transaction processing (OLTP) environments, query-intensive data warehouses and demanding internet applications. Oracle exploits clustered hardware configurations to improve performance. Strong encryption and integrity algorithms ensure data is kept safe. Oracle established the Structured Query Language (SQL) which is used to create, delete and retrieve the data stored in the database (Oracle, 2002).

2.3.2.2. MySQL

MySQL is the leading Open Source database system. It is available to download for free, and can be used without a license. It is available for many platforms, including UNIX. MySQL incorporates many of the features of the Oracle Database Server. It supports a broad sub-set of the SQL syntax, and adds its own useful functions. MySQL can be configured for a particular performance, using its independent storage engines. If transaction support is required, it uses a particular storage engine. If transaction support is not required, another storage engine is used for maximum performance. MySQL allows authenticated access, so only authorised users are able to access the database (MySQL, 2002).

2.3.2.3. Gadfly

Gadfly is an open source, relational database management system which uses a large subset of very standard SQL as its query language and Python modules and optional Python/C extension modules as its underlying engine. Gadfly stores the active database in memory, with recovery logging to a file system (Gadfly frequently asked questions, 2002). Zope has Gadfly built-in, which means it requires no third-party adaptor to use it. Gadfly is a fast database system, however since the live databases are kept in memory during active use, it is not appropriate for very large databases.

2.3.2.4. SAPDB

SAP DB is a state-of-the art, SQL-based, industrial-strength database system, created by SAP AG. SAP DB is platform-independent, so users can deploy it for a wide array of projects. SAP DB is highly scalable, easy to use and maintain, and provides full transaction support. SAP DB also goes beyond relational processing by offering object orientation as well as support for managing unstructured data. This comprehensive set of features and their availability as Open Source will benefit the entire database development community (SAP, 2003).

2.3.2.5. Selection of database technology

The database technologies described above, employ many of the same features. A matrix will be used to decide on an appropriate technology.

	Oracle	MySQL	Gadfly	SAPDB
Works on UNIX	YES	YES	YES	YES
Open source		YES	YES	YES
Large database support	YES	YES		YES
Works with Zope	YES	YES	YES	YES*

* Using an unsupported third-party adaptor

Chosen database technology: MySQL

MySQL was chosen because of four main features:

Firstly, MySQL is free to download and does not require any license to run; secondly MySQL is faster than Oracle, and can have its performance further tuned. The Operations Board does not use transactions. Therefore MySQL would be used without transaction support. This increases the performance of MySQL; thirdly it is scaleable. If the load on MySQL becomes too great, it can be used on several machines and linked together, therefore 'clustering' the systems' power; fourthly, MySQL builds on the standard SQL support with features to enhance the use of the databases.

2.4. Investigation of process models

When developing software, of any type, process models are used. Boehm (1988) stated that process models perform two vital functions. Firstly, it determines the order of the stages involved in development and evolution and secondly it establishes the transition criteria for progressing from one stage to the next. Some of the process models are detailed below.

2.4.1. Waterfall model

In 1970, Royce proposed what is now known as the Waterfall Model. The idea of this model is that it has a specific set of phases, each with entry and exit criteria. A typical set of phases are: software requirements, preliminary design, detailed design, implement and test, and integration test.

One of the strengths of the waterfall model is that it allows good customer and developer planning by having specific linear stages. A weakness of the model is the developer needing an entire set of requirements before beginning design.

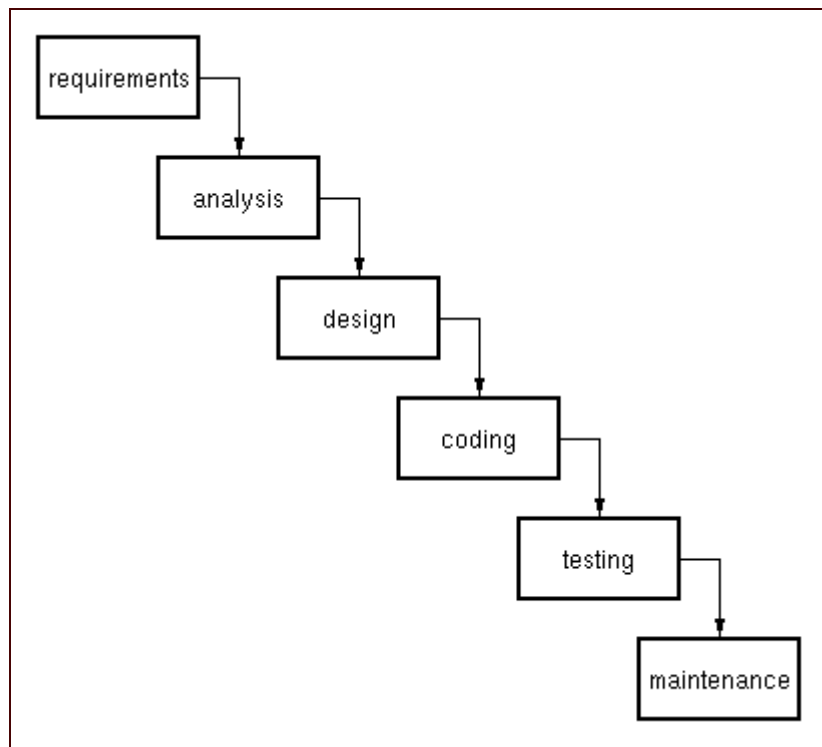


Figure 2 - The traditional waterfall model (University of Southampton, 1999)

2.4.2. Spiral model

The spiral model, proposed by Boehm in 1988, is similar to several waterfall models in a row. The initial concept spirals outwards, through all of the relevant stages to completion.

One of the strengths of the spiral model is that it provides a mechanism to cope with the fact that the user requirements typically change over the development life cycle. When requirements change, they are incorporated in the next turn of the spiral.

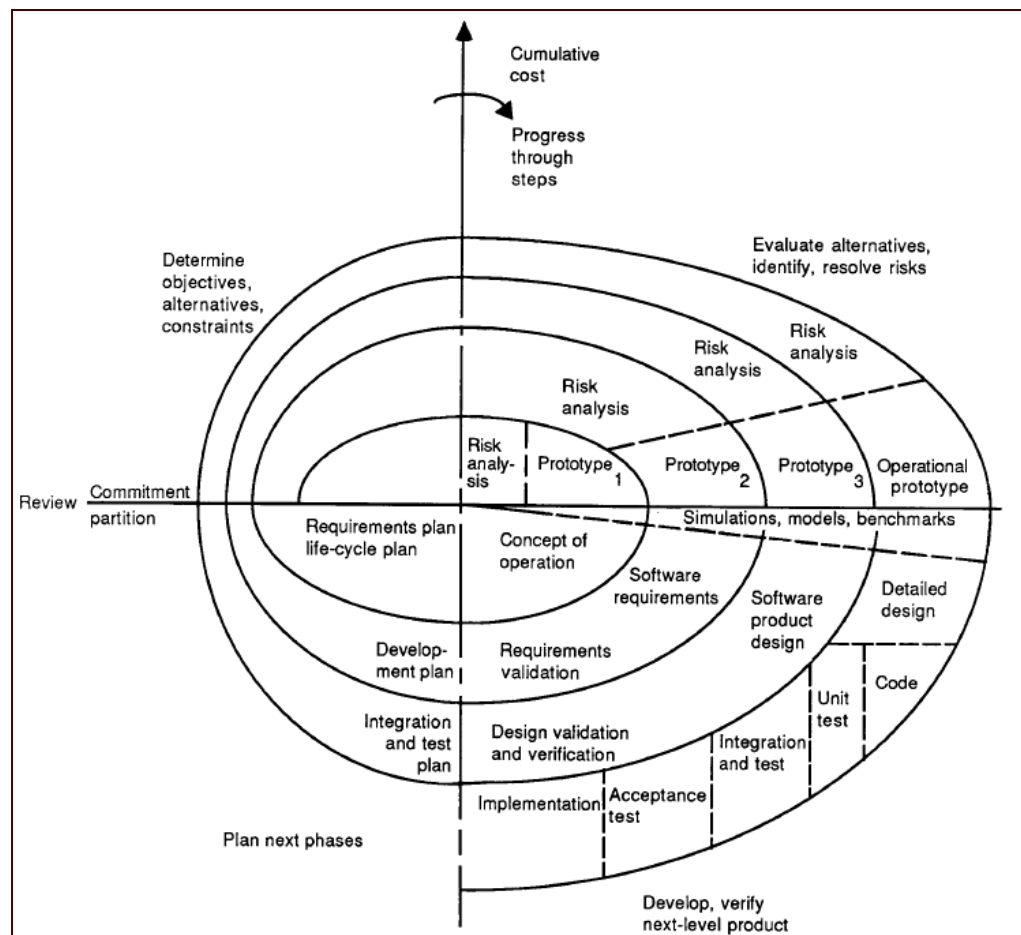


Figure 3 - The spiral model (Boehm, 1988)

2.4.3. Rapid prototyping

Rapid prototyping extensively uses prototyping to refine requirements, hence the 'prototyping' in the name. The 'rapid' part implies the use of special software tools to design interfaces quickly.

Rapid prototyping, if used in an early phase of the development life cycle, may help to set requirements, and therefore allowing an effective waterfall approach. Then in the design phase using the Builds model in parallel will further develop and refine the requirements.

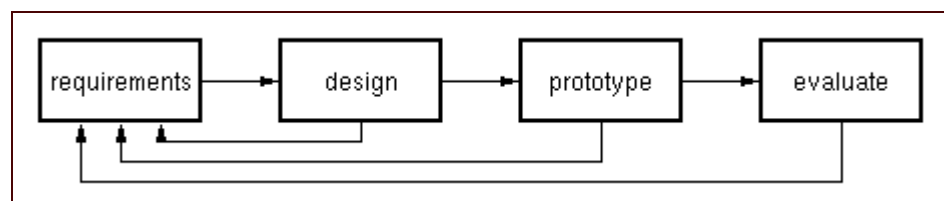


Figure 4 - The prototyping model (University of Southampton, 1999)

In rapid prototyping interactive prototypes are developed which can be quickly replaced or changed in line with design feedback. This feedback may be derived from colleagues or users as they work with the prototype to accomplish set tasks (UsabilityNet, 2003).

Rapid prototyping gives users a significant demonstration of what the system is about. It also permits the swift development of interactive software prototypes (UsabilityNet, 2003). A big benefit of prototyping is that the initial design does not have to be finalised before a prototype can be created.

2.4.4. Selection and justification of model

The Operations Board will be built using an object orientated web development system. Using this system, makes it very simple to build prototypes, then change them, and build on them later. For this reason, the design will be based on Rapid Prototyping. This will allow the end users of the Operations Board chance to use the system and suggest improvements.

2.5. Investigation of design methodologies

There are many different design methodologies. Some of these are based on other methodologies, and provide enhancement. Some of the most popular design approaches are detailed below.

2.5.1. Structured design methodology

Structured Design was synthesised by Yourdon and Constantine in 1979. The principal involves splitting the program into a set of steps, considered as 'black boxes'. These black boxes are subprograms and represent the functions required to satisfy the requirements of the user.

At the top level a complete solution is provided. At the lowest level, this is broken down into many black boxes. At both levels, the solution is provided, but it is just shown in different detail levels.

This method is extremely popular when developing programs. This is because it satisfies two key elements required when designing programs; a logical program layout and an effective mechanism for passing of parameters between subprograms.

With respect to web design, a structured design approach would involve designing the entire web site, and creating it stage after stage, in a structured order.

2.5.2. Object-oriented methodology

Object-oriented design is a design strategy where system designers think in terms of 'things' instead of operations or functions (Sommerville, 2001: 261). These 'things' are similar to 'black boxes' mentioned in the previous section.

The objects are related to the solution, but do not however provide the solution. The developer inevitably has to add new objects and to transform problem objects to implement the solution (Sommerville, 2001: 261).

Objects used within the system contain all of the relevant procedures and data, and would not be included in a system diagram. The diagram would only include the objects and how they interact with each other. This means that the developer can change the entire code and data structures within objects without changing the design. This is a big benefit.

With respect to web design, an object-oriented approach would involve assigning each web page/database interaction as an object. These objects can then be shown to interact without divulging the structure within them. Leaving them open to change.

2.5.3. User Centred Design (UCD) methodology

User centred design employs the idea of designing the software around the intended user. The approach typically entails involving users in the design and testing of the system so that their feedback can be obtained. Prototypes are usually employed to do this and designs are modified in light of the user feedback (Nectar, 2003).

Following this process to develop software can result in a number of significant advantages for the software developer, by producing software which:

- is easier to understand and use, thus reducing training costs.
- improves the quality of life of users by reducing stress and improving satisfaction.
- significantly improves the productivity and operational efficiency of individual users and consequently the organisation (Nectar, 2003).

A user centred design approach is ideal for web design. It allows the developer to implement the ideas that were described in section 1.4 about web usability. Making the web site based on the users' feedback automatically makes the web site more usable.

2.5.4. Top-down methodology

This method involves taking the problem and breaking it down into smaller problems. Each problem is then broken down until it can be solved in a single unit of programming code. This method is most applicable if the developer has an idea of what the entire system should contain, in advance of the design (Zelle, J. M et Al, 1994).

With respect to web design, this method involves creating the initial web site, and for example, creating placeholders where future pages will be created. The look and feel of the web site then can be established early on, and the rest of the site can be created based on this.

2.5.5. Bottom-up methodology

This method is most applicable if the developer is not able to grasp the full extent of the problem, or not able to initially understand it. The developer would design the individual components, and then group these together at a later date (Zelle, J. M et Al, 1994).

With respect to web design, this method is useful if the designer does not know how the entire website will function, but understands how to create the individual components. These components can be created, and then at a later stage, the main site can be created from grouping these. The look and feel can then be applied to all of the components.

2.5.6. Selection and justification of methodology

Selected design methodology: User Centred Design/Top-down

User centred design (UCD) has been chosen because it is the most relevant methodology to use with Rapid Prototyping. Each prototype created will be refined by user feedback, and it is therefore **essential** that the prototypes are designed for the user.

As well as UCD, Top-down methodology has also been chosen. This is because the initial functionality of the website is known. As the initial components of the system are also known, the developer is able to create a main site, with the components linked using the same look and feel.

3. Design of the Operations Board

The website will be made up of many objects. These include Dynamic Template Mark-up Language (DTML) pages, SQL methods, images and scripts. In order to be able to retrieve data from a database, two things need to be created; a database connection and a database.

3.1. Database/table creation

Within the database, three tables will be created: users; tasks; hosts. These tables require no relationship between them as they will not necessarily contain any related data. The initial concept was that all tasks would be based on a particular system, a host. However, over the placement year, there were many tasks that had no relevance to a host. This is the reason no relationship between the hosts and tasks table is created.

3.1.1. The 'user' table

This table will store the data about the users of the Operations Board. It will store data such as the user's full name and e-mail address. The structure of the table will be as follows:

Field name	Data type	Description
username	Variable character, maximum length 30.	The user's username, this the username they use to log into the Operations Board.
Email	Variable character, maximum length 30.	The user's e-mail address.
Firstname	Variable character, maximum length 30.	The user's first name.
secondname	Variable character, maximum length 30.	The user's second name.

The primary key for this table is *username*. The code for creating this table in MySQL is shown here:

```
CREATE TABLE users (
  username varchar(30) NOT NULL default '',
  email varchar(30) default NULL,
  firstname varchar(30) default NULL,
  secondname varchar(30) default NULL,
  PRIMARY KEY (username)
) TYPE=MyISAM;
```

3.1.2. The 'tasks' table

This table will store the data about the tasks of the Operations Board. All of the information about each task will be stored in this table. The structure of the table will be as follows:

Field name	Data type	Description
tid	Integer, maximum length 11.	The number of the task, this is automatically created by MySQL on insertion of the data.
owner	Variable character, maximum length 15.	The 'username' of the current owner of the task.
requestor	Variable character, maximum length 15.	The 'username' of the user that requested the task.
summary	Variable character, maximum length 255.	A brief summary of the task.
start_date	Date format	The date when the task started/commences.
start_time	Variable character, maximum length 5.	The time when the task started/commences.
approx_end_date	Date format	The approximate date when the task ends/ended.
approx_end_time	Variable character, maximum length 5.	The approximate time when the task ends/ended.
priority	Fixed character of	The priority of the task, for example P1

	length 2.	for top priority tasks.
ops_status	Variable character, maximum length 15.	What type of task this is at present, for example, unplanned means an unplanned outage.
status	Variable character, maximum length 15.	The current status of the task. Examples include: 'In progress', 'Cancelled' and 'Closed'.
country	Variable character, maximum length 15.	The country that the host is located in (if relevant).
site	Fixed character of length 5.	The site that the host is located in (if relevant). Examples include: MUC07 for Munich or BER02 for Berlin.
serialnum	Variable character, maximum length 15.	The serial number of the host (if relevant).
hostname	Variable character, maximum length 30.	The name of the host (if relevant).
macaddr	Variable character, maximum length 17.	The Ethernet (or MAC) address of the host (if relevant).
model	Variable character, maximum length 30.	The model of the host (if relevant).
operations_manager	Variable character, maximum length 50.	The name and/or details of the operations manager supervising the task.
task_type	Variable character, maximum length 30.	The actual type of the task to be carried out. Examples include: 'Software Upgrades' and 'Hardware Maintenance'.
description	Longtext format.	The description of the task as entered by the owner.
notes	Longtext format.	Notes that have been added to the task.
sd_number	Variable character, maximum length 10.	The Service Desk ticket number that this task is based on (if relevant).
gcn_number	Variable character, maximum length 10.	The Global Change Notification (GCN) number that this task relates to (if relevant).
radiance_number	Variable character, maximum length 10.	The Radiance number related to this task (if relevant).

The primary key for this table is *tid*. The code for creating this table in MySQL is shown here:

```
CREATE TABLE tasks (  
    tid int(11) NOT NULL auto_increment,  
    owner varchar(15) default NULL,  
    requestor varchar(15) default NULL,  
    summary varchar(255) default NULL,  
    start_date date default NULL,  
    start_time varchar(5) default NULL,  
    approx_end_date date default NULL,  
    approx_end_time varchar(5) default NULL,  
    priority char(2) default NULL,  
    ops_status varchar(15) default NULL,  
    status varchar(15) default NULL,  
    country varchar(15) default NULL,  
    site char(5) default NULL,  
    serialnum varchar(15) default NULL,  
    hostname varchar(30) default NULL,  
    macaddr varchar(17) default NULL,  
    model varchar(30) default NULL,  
    operations_manager varchar(50) default NULL,  
    task_type varchar(30) default NULL,  
    description longtext,  
    notes longtext,  
    sd_number varchar(10) default NULL,  
    gcn_number varchar(10) default NULL,  
    radiance_number varchar(10) default NULL,  
    PRIMARY KEY (tid)  
) TYPE=MyISAM;
```

3.1.3. The 'hosts' table

The hosts table will include the data about each of the systems that IT Operations currently supports. This data will be retrieved from a system called Registration Tool. This will be inserted into the hosts table using a feature of the Operations Board that will be described later. Some of the data will not be used by the Operations Board at present, and are shown in

italics. These are included in the table in order to allow for future expansion of the system.

Field name	Data type	Description
hid	Integer of maximum length 11.	A unique identification number for the host.
hname	Variable character, maximum length 20.	The name of the host.
hcountry	Variable character, maximum length 15.	The country that the host is located in (if relevant).
hmac	Variable character, maximum length 17.	The Ethernet (or MAC) address of the host (if relevant).
hsite	Variable character, maximum length 10.	The site that the host is located in (if relevant). Examples include: MUC07 for Munich or BER02 for Berlin.
hmodel	Variable character, maximum length 15.	The model of the host.
<i>huser</i>	Variable character, maximum length 15.	The user of the host.
<i>howner</i>	Variable character, maximum length 15.	The owner of the host.
<i>hdept</i>	Variable character, maximum length 15.	The department that owns the host.
<i>hdiv</i>	Variable character, maximum length 50.	The division that the host is contained in.
hserialnum	Variable character, maximum length 15.	The serial number of the host.
<i>hclass</i>	Variable character, maximum length 30.	Any relevant information about the host.

The primary key for this table is *hid*. The code for creating this table in MySQL is shown here:

```
CREATE TABLE hosts (  
  hid int(11) NOT NULL auto_increment,  
  hname varchar(20) NOT NULL default '',  
  hcountry varchar(15) default NULL,  
  hmac varchar(17) default NULL,  
  hsite varchar(10) default NULL,  
  hmodel varchar(15) default NULL,  
  huser varchar(15) default NULL,  
  howner varchar(15) default NULL,  
  hdept varchar(15) default NULL,  
  hdiv varchar(50) default NULL,  
  hserialnum varchar(15) default NULL,  
  hclass varchar(30) default NULL,  
  PRIMARY KEY (hid)  
) TYPE=MyISAM;
```

The website can now include database content. The database functionality is added to Zope using two objects; a database connection and SQL methods. A database connection provides the access to the database, with the authentication needed. SQL methods are objects that use the database connection to carry out the SQL code that has been entered in them.

An example of an SQL method used in the Operations Board can be found in Appendix A2.

Every single page within the Operations Board will use SQL methods to retrieve data and incorporate this data within the web pages.

3.2. Interface/GUI creation

3.2.1. Security

The Operations Board will need password protected authentication. This is simple with Zope as it allows you to define a list of users, and their

passwords within a directory. A user then acquires the same permissions for each directory followed.

Zope will automatically prompt the user for their username and password when a page is requested which requires authentication. When the user has authenticated, the security system within Zope defines a variable within its Security Manager. This variable will allow data to be retrieved from the database dependant on the current user, which is detailed later.

3.2.2. Look and Feel

Sun Microsystems web pages are designed around a particular theme, which uses corporate colours and images. The look and feel of the corporate image is applied to web sites using templates. These templates are available for download at <http://www.sun.com/smc/web>. The figure below shows a sample web page taken from the templates.

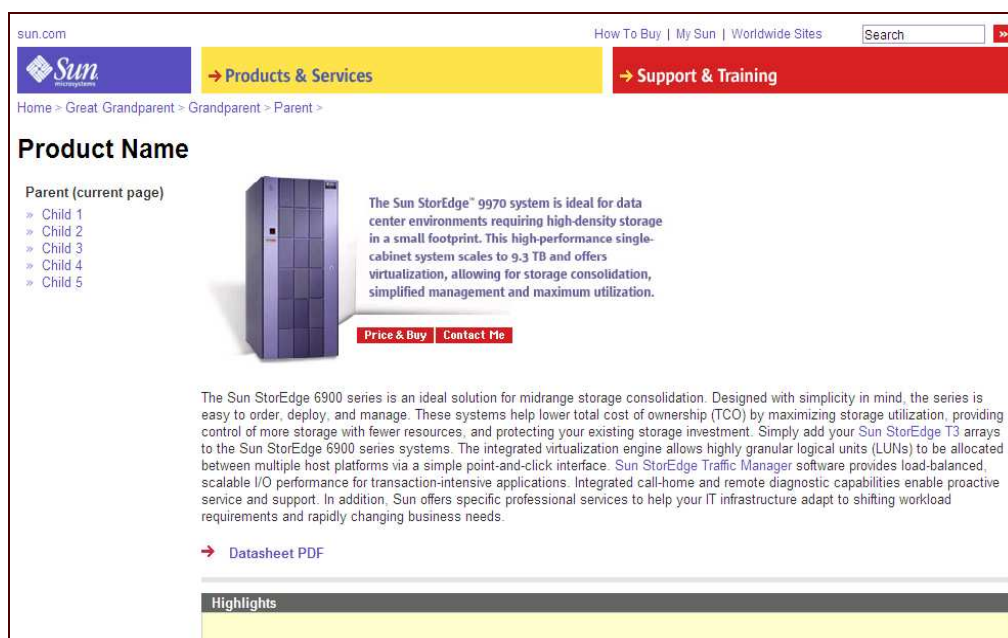


Figure 5 - The standard look and feel of a Sun web page

The Operations Board will be based on these same templates. However, different parts of different templates will be added together to make the page complete. A page from the Operations Board is shown below. Each of the different components is numbered, and then explained afterwards.

Controlled Access Area Logged in as: Ben Mason (bm128199)

Sun → Products & Services 1 → Support & Training

World Wide Client Services Operations Board Launch WHITEBOARD

Adding Tasks
Add task
Add MULTI tasks

Add task by hostname:
[input] [button]

Browse Tasks
My tasks
Unassigned tasks
All tasks
Browse tasks by site
Advanced search

Search: 2 [input] [button]

Search task ID: [input] [button]

More information

Admin
Change my details
Add user
View/Edit hosts
Bulk upload

FEEDBACK

All tasks

	Owner	Start date	Priority	Summary	Tools
<input type="checkbox"/>	Uwe Schrader	2003/01/14	P1	File scanning persistent DIMM error ...	[icon] [icon] [icon]
<input type="checkbox"/>	Naznin Todiwal	2003/02/04	P1	check backup	[icon] [icon] [icon]
<input type="checkbox"/>	Daniel Narkis	2003/02/05	P1	check sunray for overheating	[icon] [icon] [icon]
<input type="checkbox"/>	AT&T	2003/02/05	P1	SWAN HUB QOS ROUTER UPGRADE	[icon] [icon] [icon]
<input type="checkbox"/>	Naznin Todiwal	2003/02/09	P1	3	[icon] [icon] [icon]
<input type="checkbox"/>	AT&T	2003/02/13	P1	SWISS SWAN HUB ROUTER UPGRADE	[icon] [icon] [icon]
<input type="checkbox"/>	Unassigned	2003/05/15	P1	Upgrade memory in mail server	[icon] [icon] [icon]
<input type="checkbox"/>	Uwe Schrader	2003/01/17	P2	hastat looks inconsistent	[icon] [icon] [icon]
<input type="checkbox"/>	Frank Nordbeck	2003/02/13	P2	sunray server has crashed	[icon] [icon] [icon]
<input type="checkbox"/>	Mark Gifford	2003/02/10	P3	stop printsv spamming dutyman	[icon] [icon] [icon]

Close » Tasks 1 to 10 (of 11 tasks) Pages: 1 - 2 | » Next

Figure 6 - The 'All Tasks' screen in the Operations Board. This is where the user can see all of the open tasks.

1 Controlled Access Area and masthead

This grey banner signifies that the area the user is in a controlled access area. This is a requirement from Sun. The Sun web resources page states “When users are in an area that restricts access to general users (e.g. password-protected, extranet), the Secure Area Header appears, to remind the user that they are in a restricted access area of the site.”

This masthead identifies the web site and anchors the page. It serves as top-level navigation. Clicking on the Sun logo takes the user to <http://www.sun.com> and clicking on the Products or Support takes the user to the product and support pages respectively.

Both the Controlled Access Area banner and the masthead do not change during the use of the Operations Board.

2 Side navigation bar

The side navigation bar is designed using the Sun colours. However it is not based on any template. This is because it uses search boxes and images, which are not part of any of the Sun templates, but required by the Operations Board. The side navigation bar remains the same throughout the use of the Board.

3 Main body

This shows the title of the current page so the user knows exactly what they are doing, or what has been done.

The data in the table is formatted using the Sun standard colours. An example of a table from the Sun templates is shown here:

Standard
Featured Content [\[code and specification\]](#)





G1 Featured Content Component Title		
Feature	Function	Benefit
primum; nec, si miserum Fortuna Sinonem finxit, uanum etiam	Thymoetes duci intra muros hortatur et arce locari	quorum melior sententia menti, aut pelago Danaum insidias
quidquid id est, timeo Danaos et dona ferentis. sic fatus ualidis ingentem	accipere? aut quid iam misero mihi denique restat, cui neque apud Danaos usquam locus, et super ipsi Dardanidae infensi	Capys, et quorum melior sententia menti, aut pelago Danaum
primum; nec, si miserum Fortuna Sinonem finxit, uanum etiam	Thymoetes duci intra muros hortatur et arce locari	quorum melior sententia menti, aut pelago Danaum insidias
quidquid id est, timeo Danaos et dona ferentis. sic fatus ualidis ingentem	accipere? aut quid iam misero mihi denique restat, cui neque apud Danaos usquam locus, et super ipsi Dardanidae infensi	Capys, et quorum melior sententia menti, aut pelago Danaum

[» See detailed Features and Benefits](#)

Figure 7 - The standard Sun colours implemented in a sample table

The flag represents the country associated with the task. All tasks are sorted by priority, and start date.

The icons shown on the right of the table are the tools available to the user for each task. Each of the icons and their functions is detailed below.

	Update the task - the user is able to load and update this task.
	View the task - shows the user detailed information about this task.
	Add a note - allows the user to view, and add notes. The user is unable to delete any notes.
	Clone task - allows the user to create another task based on the data within this task.

Below the formatted table is a series of pagination links. These links allow the user to quickly skip through pages of tasks to find the task they need. The page links change dynamically depending on how many tasks are currently showing. The pagination has also been styled using Sun templates. However the Close Task button is an addition solely for the Operations Board. An example of different pagination styles is shown below.

<p>Standard Pagination [code and specification]</p> <p>Page 12 (110-120 of 546 results) Pages: Previous < 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 > Next</p>
<p>Variation 1 First as Current [code and specification]</p> <p>Page 1 (1-10 of 546 results) Pages: 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 > Next</p>
<p>Variation 2 Last as Current [code and specification]</p> <p>Page 10 (90-98 of 98 results) Pages: Previous < 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10</p>

Figure 8 - The standard sun pagination design templates.

The Close button allows the user to select one, or many tasks shown and close them. This function automatically sets the close time of the task and sets its status to closed.

3.2.3. Code modules

As mentioned before, every web page of the Operations Board will use Document Template Mark-up Language (DTML). DTML allows data to be retrieved and manipulated. The data is retrieved from the database using SQL methods, this is then parsed by DTML code. Here is an example of some DTML code:

```
<dtml-if "color==red">
  The colour is Red
<dtml-else>
  The colour is not Red
</dtml-if>
```

This code will display **The colour is Red** if the variable color is set to 'red'. In all other cases, it displays **The colour is not Red**. The dynamic information in the Operations Board is displayed in the same way. On a page that displays user information `<dtml-var firstname>` will display the user's first name.

An example of DTML code used in the Operations Board can be found in Appendix A3.

The Operations Board is made up of many different functions and tools. The functions accessible by the side bar are described below.


Add task

This adds a task to the Operations Board. The user is prompted for the host name, or they may select no host if the task is not related to a host.


Add MULTI tasks

If many similar operations are to be performed on several systems, then the user can create a MULTI task. This gives the user the same options as a standard task. However it allows the user to select more than one host.


Add task by hostname

This is a quick way of getting to the host look-up screen. The host name is entered in this box and then the user clicks on '>>'.



My tasks

This function displays in tabular form all the tasks which are open, and are owned by the current user. The user is then presented with a set of tools for each task.


All tasks

This displays all tasks that are presently open by users of the Operations Board.



Browse tasks by site

This presents the user with a list of countries, dynamically created from the database. The user then selects a country. The subsequent page prompts the user with a list of sites based in the country they have chosen. A list of open tasks that are based on hosts in the selected site is displayed.


Advanced Search

This presents the user with selection lists and input boxes to perform searches on all tasks.


Search hostname and Search task ID

These search boxes simply search all tasks with the criteria entered. Either or both fields can have criteria entered.


Change my details

Allows the user to change their first and last name, e-mail address and password. These changes are only effective in the Operations Board.

Add User

A new user can be added to the Operations Board using this function. A username, first and last name, e-mail address and password are needed to create a new user.

View/Edit hosts

The host information can be viewed and edited within the Operations Board. When the hosts are displayed, a set of tools are displayed, similar to those of the tasks.

Bulk upload

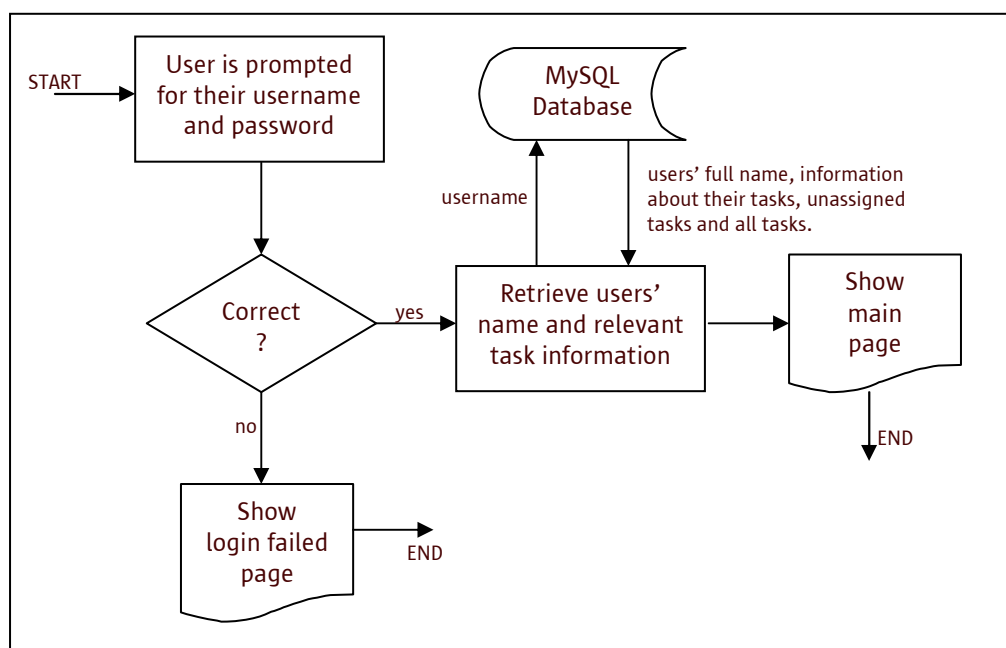
This feature allows a bulk upload of host information. Sun Microsystems uses a tool called Registration Tool or RegTool. RegTool produces Comma Separated Value (CSV) text files containing the host information for a region. The Operations Board has been designed to import this data directly and insert it intelligently into the host table.

Feedback

This link allows the user to post feedback to the webmaster of the European region if there are any problems with the board, for example. This has enabled rapid implementation of changes in the Operations Board.

The tools that are accesible from the task listings were described in section **3.2.2**. Some of the main components of the Operations Board are detailed below in flow charts to show how the components work.

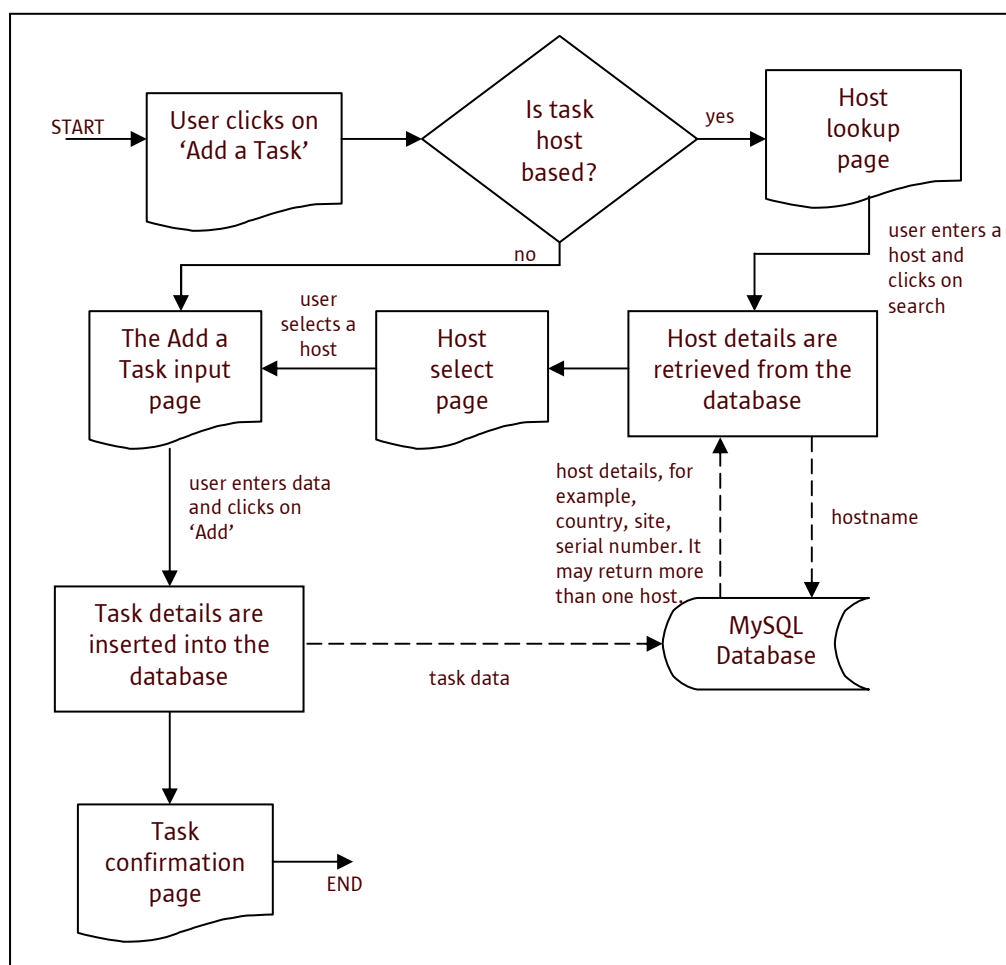
3.2.3.1. The Main page



Description of the stages:

- The user is prompted for their username and password.
- If the username and password combination is incorrect, the *authentication failed* page is shown.
- An SQL method is used to call user and task information from the database. This is then dynamically coded into the web page before the web server sends it to the user.

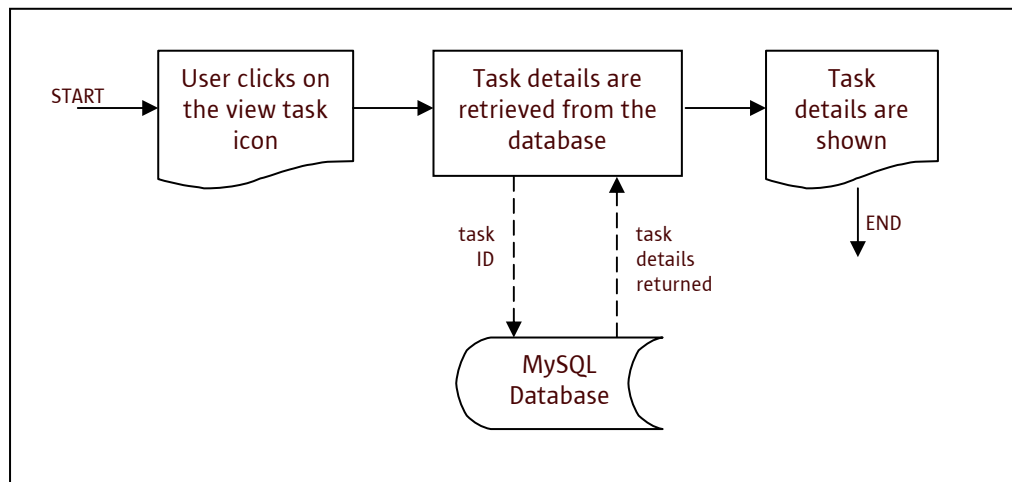
3.2.3.2. Adding a Task



Description of the stages:

- The user clicks on *Add a task* from the side bar.
- The host look-up page is displayed. If the task is not host related, then the user is able to go directly to the data input page.
- The user now enters a hostname and clicks on search.
- The next page is populated with host data from the database.
- The user selects a host.
- The Add a task input page is shown, and if it is host based, then the host information will be filled in.
- The user adds the relevant data and clicks on add.
- An SQL method pushes the data into the database.
- A task confirmation page is then displayed.

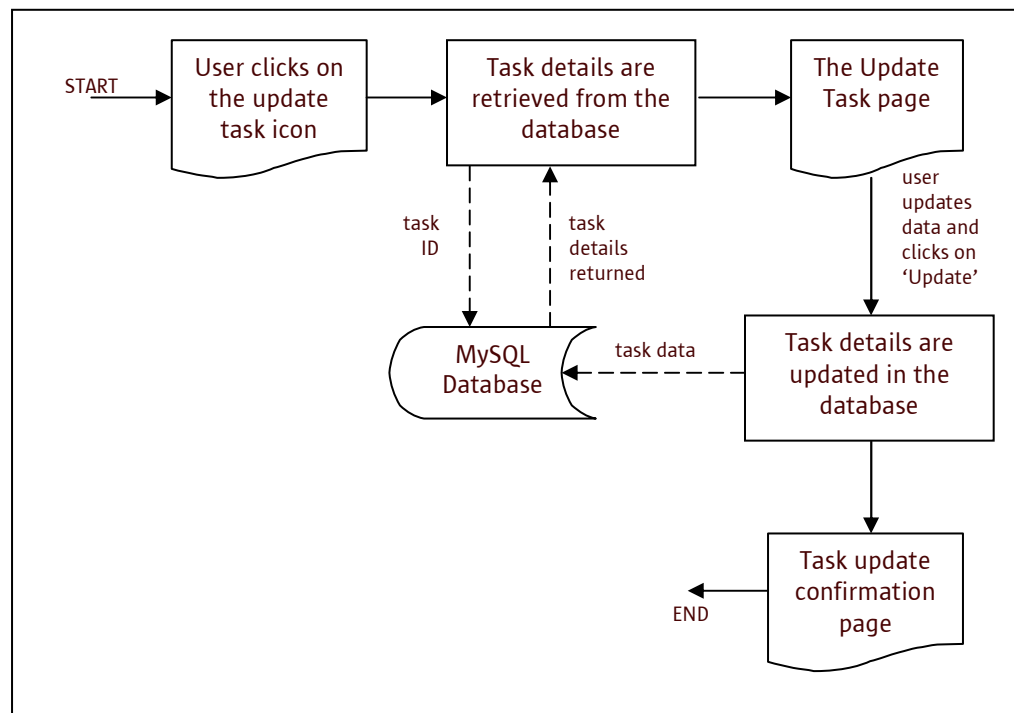
3.2.3.3. Viewing a Task



Description of the stages:

- From any of the pages showing the *view task* icon (as described earlier) the user clicks on the relevant *view task* icon.
- An SQL method retrieves the relevant data from the database based on the task ID.
- The View task page is displayed with the information from the database shown.

3.2.3.4. Updating a Task



Description of the stages:

- From any of the pages showing the *update task* icon (as described earlier) the user clicks on the relevant *update task* icon.
- An SQL method retrieves the relevant data from the database based on the task ID.
- The update task page is populated with the task data. This page resembles a form with input boxes and selection lists.
- The user then changes/inserts the new data and clicks on *Update*.
- This data is then updated in the database.
- A task update confirmation page is shown.

3.2.3.5. Searching for a Task

When a user searches for a task, there are two alternatives available. Either they can search by task ID and/or hostname, or they can use the advanced search to enter more criteria. Both options use the same SQL method, so it is more simple.

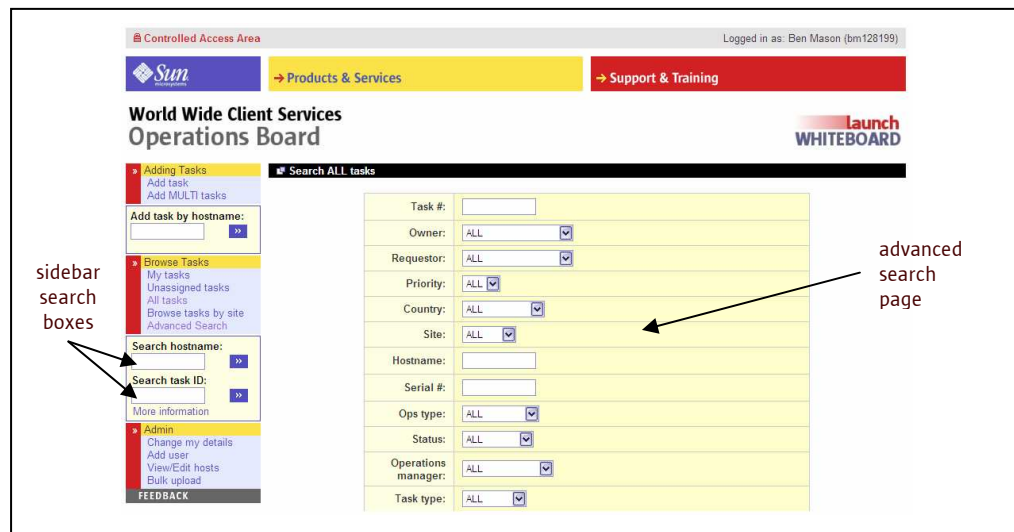


Figure 9 - A screen shot showing both the sidebar search (on the left) and the main advanced search fields (in the main body).

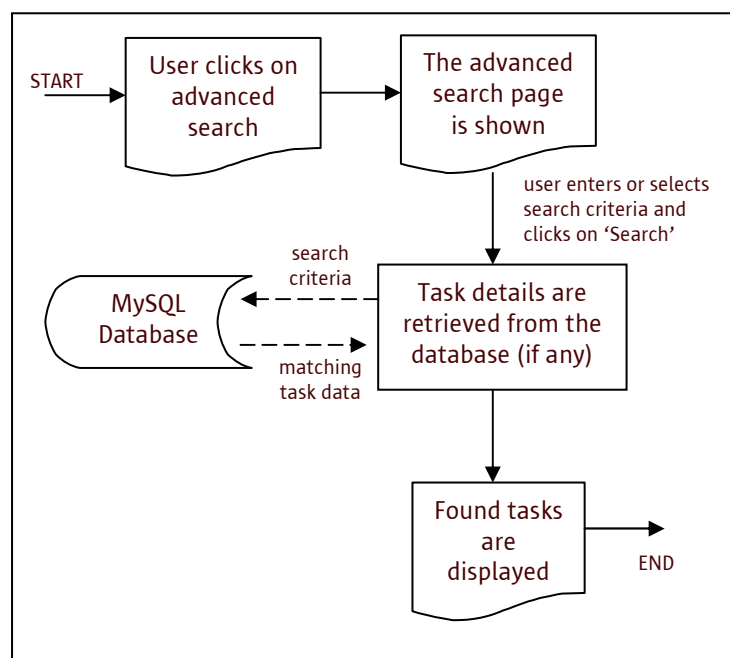


Figure 10 - How searching works when the user selects to use the advanced search page.

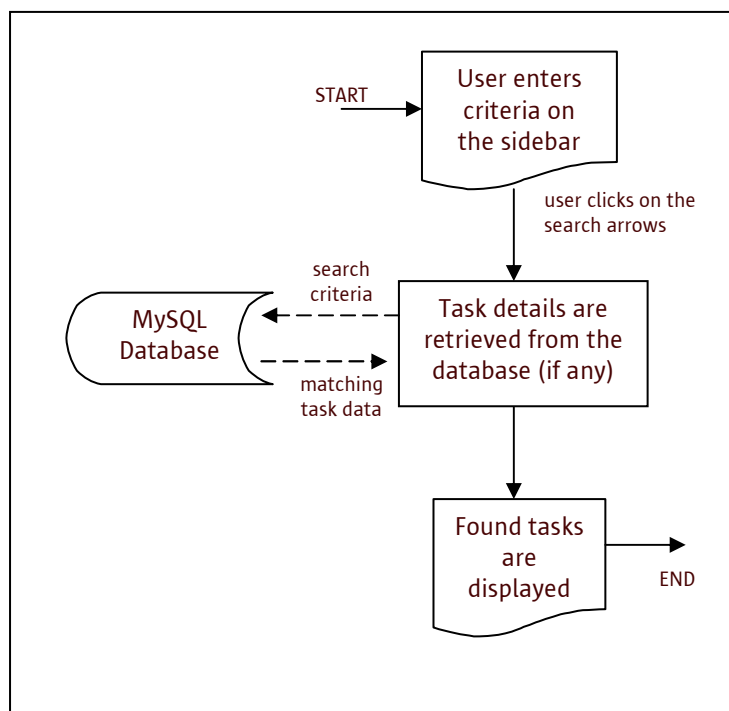


Figure 11 - How searching works when the user enters search criteria on the side bar.

Description of the stages:

- The user either enters criteria directly from the side bar or from the advanced search page.
- An SQL method retrieves data from the database based on the search criteria entered, and returns any tasks it finds.
- These tasks are then parsed into tables (like shown earlier) with the tools to update and view them.

3.2.4. The Whiteboard

The Whiteboard is to become the **direct** replacement for the existing physical whiteboard. It will display the same four pieces of information: the planned outages; the unplanned outages; the on-going maintenance; the maintenance that was completed less than twenty four hours ago.

Planned outages

This section will display all open tasks that have the status 'planned'. Any tasks that started more than twenty four hours ago are highlighted in red.

Unplanned outages

This section will display all open tasks that have the status 'unplanned'. Any tasks that started more than twenty four hours ago are highlighted in red.

On-going maintenance

This section displays any task that is open and is not a planned or unplanned outage.


Recently completed maintenance


This section displays any task that is open and is not a planned or unplanned outage closed less than twenty four hours previously.






All of the sections are ordered by priority and then date order. Each section will display 50 tasks and if a task is clicked on, it will open it up in another browser window. There is one main reason for this: users may want to use the Whiteboard on their desktops so they must be able to load tasks into their browser from it.

The Whiteboard will use frames in a 2x2 formation. Planned and unplanned outages at the top and the maintenance sections at the bottom. These frames will be automatically refreshed, every two minutes, by using a standard HTML header like the one below.

```
<META HTTP-EQUIV="refresh" content="120;URL=whiteboard_unplanned">
```

1 Unplanned Outages					
	Started	Ends	Host	GCN	
	2003/03/29 14:21	2003/03/31 23:13	sr-ezrh04-02	none	P2
Frank Nordbeck		sunray server has crashed			

1 Planned Outages					
	Started	Ends	Host	GCN	
	2003/02/13 14:08	2003/03/26 14:08	none	none	P1
AT&T		SWISS SWAN HUB ROUTER UPGRADE			

5 On-going Maintenance					
	Started	Host	GCN	Owner	Summary
	2003/01/14	sr-egra02-02	none	P1 Uwe Schrader	File scanning persistent DIMM error matches
	2003/01/17	emuc07-cluster-2	none	P2 Uwe Schrader	hastat looks inconsistent
	2003/02/04	ibis	none	P1 Naznin Todiwal	check backup
	2003/02/05	turkeysrv	none	P1 Daniel Narkis	check sunray for overheating
	2003/02/09	hoffa	none	P1 Naznin Todiwal	none



2 Recently completed maintenance					
	Started	Host	GCN	Owner	Summary
	2003/02/13	sr-ezrh04-02	none	P2 Frank Nordbeck	sunray server has crashed
	2003/02/10	printsrv	none	P3 Mark Gifford	stop printsrv spamming dutyman

Figure 12 - The Whiteboard - Two of the tasks are overdue and are highlighted in red.

3.2.5. Constraints

Some of the data used in the Operations Board is actively processed by the board itself. For example, the Whiteboard uses the status field to determine whether an outage is planned or unplanned. For reasons like that, it is important that constraints are used.

When adding or updating a task on some of the fields, a user is presented with a drop down list. For example, in the status field there are three options: planned outage; unplanned outage; maintenance. The user then has to select one of these. The default selection is *maintenance*. This prevents errors in the database such as a user typing ‘unplanned maintenance’.

The fields which have constraints and the reasons for them are detailed below. Implementing these constraints are selection lists, which only allow the user to select one of the choices.

Owner

Each user in the system has a unique *username*. However the *username* is usually an unwieldy string (eg. bm128199). This is not easy to remember so each *username* has an associated *friendly name* (eg. Ben Mason). Both *usernames* and associated *friendly names* are stored in the database. A drop-down owner list is generated automatically by using An SQL method. Selecting a *friendly name* from the drop-down list automatically substitutes the *username* as the owner of the current task.

Requestor

The requestor field is always set to the *username* of the person who originally created the task. The requestor field **cannot** subsequently be changed.

Priority

The priority can only be one of four possible options: P1; P2; P3; P4.

Operations Type

This can be only one of three options as discussed earlier: planned outage; unplanned outage; maintenance.

Status

Status can only be one of a set of options: 'proposal'; 'assigned'; 'in progress'; 'pending'; 'issue'; 'on hold'; 'completed'; 'closed'; 'cancelled'.

Manager

Manager is the name and email address of the manager in charge of the region of operations. Only one of a set of managers can be chosen, therefore using a list is the best approach.

Task Type

Task Type is also a set list of options. These options include: 'hardware'; 'software'; 'network'; 'utility'; 'other'. Task type refers to the actual type of maintenance carried out during the task.

Manager, Task Type, Status, Operations Type and Priority have not been hard coded into the Operations Board. They exist as a simple list within the Zope management interface which can be edited. The changes will be effective immediately.

4. Development

4.1. Implementation of the Operations Board

Direct access to the Sun Microsystems network was not feasible for a non-employee, so the Operations Board was designed externally. The board was hosted on a Sun Blade 100 (a system that Sun produce) and connected to the internet through a fast cable connection.

The two tools installed are Zope and MySQL. Both of these packages are available for the Solaris platform, which is the platform run on the Sun Blade 100. The packages installed and were ready to work without any further configuration. A problem occurred when trying to get Zope to communicate with MySQL.

As mentioned above, a third-party *product* has to be used with Zope in order to allow it to function with MySQL. After several attempts, and help from the enormous Zope community, the *product* was installed and configured appropriately. Database access was now possible within Zope.

The DTML pages and SQL methods were created as described in the design section. Due to the fact the Operations Board would use the same masthead banner and side bar, only one copy of these DTML documents needs to be created. They can be inserted using a DTML variable such as `<dtml-var html_header>`.

Below is an example of some DTML code used within the Operations Board. Another example can be found in Appendix A3. The following code retrieves the user's full name from the user table in the database.

```
<dtml-in SQL_get_user_details>
<dtml-if "username==_.SecurityGetUser().getUserName()">
<dtml-var firstname> <dtml-var secondname> (<dtml-var
username>)</font></b>
</dtml-if>
</dtml-in>
```

The “`_.SecurityGetUser().getUserName()`” is a built-in function of Zope which returns the *username* of the current user. The effect of this code on the Operations Board is shown below.



Figure 13 - The effect of the above DTML code on the Operations Board. The snapshot shown is on the top right of every Operations Board's page.

The first version of the Operations Board was enabled on the developer's private system and made accessible to Sun employees via the web. New users were created in the Operations Board for the students currently on placement and Mervyn Sloan, the Technical Advisor for the region. These new users were asked to try out the tool and make suggestions.

After a period of initial testing, the go-ahead was given for the Operations Board to be made operational. For data security reasons the entire Operations Board package was transferred to the Sun internal network. Zope's export/import feature was very helpful in the transfer of data. MySQL also has a similar easy import/export feature.

Development of the Operations Board continued on the external private system with only individual modified components needing to be transferred

to the Sun internal network. The main core components and database would remain unchanged.

4.2. Validation and verification (V&V)

According to Boehm (1979), the difference between validation and verification is succinctly expressed:

- ‘Validation: Are we building the right product?’
- ‘Verification: Are we building the product right?’

These definitions tell us that the role of verification involves checking that the software conforms to its specification. Validation, however, is a more general process. You should ensure that the software meets the expectations of the customer (Sommerville, 2001:420).

4.2.1. Validation - Are we building the correct product?

The Operations Board was designed using the User Centred Design methodology. This approach enforced validation at every step of the design process. After the initial design and implementation, the users initiated future prototypes through feedback. This approach ensures that the new features of the Operations Board are ones desired by the users and not the developer. Unused features were removed, new features created.

An example of a feature that was removed was the ability to respond to e-mails from the system. Originally, the system would e-mail the entire Operations Team when an **unassigned** task was added. Users were able to reply to the e-mail to become the owner, and if the task were already owned by the time they had replied, they received an email informing them of this. This feature was removed due to user request. The Operations Team already

receives many e-mails and they did not particularly wish to receive even more.

4.2.2. Verification - Are we building the product right?

After initial design, the Operations Board was put into LIVE testing by the Operations team in Germany. It was run in parallel with the existing system. Any bugs in the system would be found easily, as nearly every function would be used on a day-to-day basis. These bugs would then be fixed and a new version created.

An example of a problem a user encountered was using the Close Task option on any of the tabular views of tasks. If the user selected **more** than one task and clicked on Close, the tool functioned correctly. However, if the user selected **one** task and click on Close, an error occurred. This was due to the code in the SQL methods depending on more than one input. This was changed and implemented in a new version.

4.3. User feedback

User feedback was a crucial element in both the design and implementations of the Operations Board. Some of the features that were created due to user feedback are detailed below. User feedback was also essential in finding code errors and other such mistakes.

Whiteboard frame sizes

During the design the whiteboard was set to use a 2x2 layout, allowing 50% of the screen for the outages and 50% for the maintenance tasks. After consultation with the users, the outages were given less room, as there are always more maintenance tasks than outages.

Link on Advanced Search

The Advanced Search page is where a user is able to search all tasks for specific criteria. Links were added to the search page to make certain popular searches quicker. The Operations Team has a meeting every Monday morning, for this meeting, they need to quickly see the open tasks, tasks that have been proposed, and tasks that have issues. This is done using a single click from the Advanced Search page.

Emails go to site owners

Each senior member of IT Operations is responsible for one or more production offices throughout the region. These members become the site owners of the sites they are responsible for. The owner of a task is notified whenever one of their tasks is modified. They are also notified when a new task is created for which they are the owner. As well as this, e-mail notification to the owner of the site related to the task, was requested. When a host based task is modified, or added, the owner of the site is e-mailed to inform them of the changes.

View Tasks by Host

It was requested that on the View/Edit Hosts page, a link be added so that a user may quickly browse all tasks by a specific host. This is now done by clicking on a hostname anywhere within the Operations Board. This added functionality was more desirable for the users, than just one page with the links.

Users added to Operations Board automatically

Sun Microsystems uses a centralised database for usernames, passwords e-mail addressed and other personal data. Using this database, the Operations Board is updated with member information on a regular basis. This removes the need for the *Add a User* feature. However it is left available. This is because it may be necessary to add users who do not have a centralised Sun account (eg. people with visitor accounts).

5. Reflections

5.1. Completeness of the Operations Board

The Operations Board was originally conceived as an online replacement for the existing whiteboard. During its design, the Operations Board became a personal portal for users' tasks. The online whiteboard simply became an add-on to the board, updating automatically.

During design and prototyping, requests for new features or modifications were received from the users. These features were promptly implemented into the Operations Board and then a new version was transferred to the Sun network.

The Operations Board performs every function of the existing whiteboard / e-mail system. It also performs many more useful features that were done manually, by e-mail, before the board was implemented. An example is that new maintenance tasks were written on the whiteboard then an Operations Team member would put their initials next to the task they were dealing with it. With the whiteboard in the Munich office, and Team members being in Berlin and Geneva, it was difficult for them to see 'who was doing what'.

5.2. Strengths and weaknesses of the Operations Board

In the following sections, the strengths and weaknesses of both the Operations Board and the way it was designed are discussed. These sections have been combined, as nearly every weakness and every strength of the Operations Board is due to the way it was developed.

5.2.1. Strengths

Shortly after the implementation of the Operations Board, the IT Operations department merged with similar offices in a further 12 countries. This meant that the Operations Board would now be dealing with tasks from 15 countries and handling four times the amount of tasks and outages. The Operations Board was built to be totally dynamic. There is very little data in the Operations Board that can not be changed quickly, or is not updated automatically. When the regions merged, the Operations Board was updated with the data from the hosts from the other 12 countries. The new sites, countries and hosts were now present throughout all functions. The Operations Board continues to be just as fast and just as functional as it was before addition of the earlier 12 countries.

The Operations Board is used as the primary task tracking tool. There are no tasks performed by IT Operations that are not entered into the board. During meetings, including the Monday morning Operations Team meeting, the Operations Board is used. It has since become a 'mission critical' tool. This means that the department can not work if the Operations Board is not working. A member of the Operations Team has the duty of making sure the Operations Board is functioning correctly. The Operations Board has not yet failed to work or caused any problems.

The Operations Board is fast. It is hosted on a lightly loaded departmental web server. This allows the dynamic web pages to be served to the users quickly and efficiently. The efficient way in which Zope works with MySQL also increases the speed of using the Operations Board.

The Operations Board was can be modified quickly and efficiently. The time between a user requesting an amendment and that being implemented, was sometimes only a matter of hours. The SQL code was designed to be usable in many different pages.

5.2.2. Weaknesses

When the Operations Board was first designed, the host information for tasks was entered manually. Soon, the host information was included in the Operations Board database. This meant that the tables were designed with no relationships. This was not changed during design as not every task is host-based. However there should be a relationship between the tables. At present, when the tool shows the information for a task, it has to make a separate call to look up the username in the users table. However, if there were a relationship, this would be done automatically.

As mentioned earlier, host information has to be retrieved from RegTool. This data is then imported into the Operations Board. This means a user has to extract the host information from RegTool then import it using the Bulk Upload facility in the Operations Board. This information could probably be extracted from RegTool and imported into the Operations Board automatically. This was not done during implementation because of the infrequent updates of the host information.

Originally the Operations Board was to feature a management section. This section would be available to certain users only. These users would be able to reassign, edit or close **anyone's** tasks. During the design, the Operations Manager requested that any user should be able to reassign, edit or close anyone's tasks. This means that there has to be a level of trust when using the Operations Board, as a user could reassign an undesirable task. This was changed due to the fact that the person primarily assigning tasks is the Duty Manager. However the Duty Manager can be anyone of a number of people.

5.3. Further tasks to enhance the Operations Board

The tables within the Operations Board should really be related. The owner and requestor of a task will match usernames in the users table, therefore relating the two tables. Also, the host in a task will match the host in the hosts table, therefore these can be related too. This would eradicate repeating data in the tasks table such as site, country and mac address, as these would be retrieved from the hosts table.

The Operations Board should be able to extract the data from RegTool and import it, automatically. There is a web interface to RegTool from which the data could be extracted using a text based web utility and then imported straight into the Operations Board. This could be performed as often as the Operations Team sees fit. This eliminates the need for the Bulk Upload facility.

Although the management interface was discarded during design, the need for it is apparent. The management access can easily be delegated to a different user when it is their turn to perform the Duty Manager role. Users would only be able to update their tasks, add notes and set the status to completed. The tasks would then be closed by the Duty Manager during the weekly Operations Meeting. Only the Duty Manager would therefore be able to create tasks and assign them to users. At present, any user can create a task, however its status is set to *proposal* until it is accepted by the Duty Manager.

E-mail notification of overdue tasks would be a useful feature. A user would get a daily e-mail informing them that one of their open tasks is overdue, therefore encouraging them to take a look at it. The Duty Manager would also be informed of these tasks. At present, the overdue tasks appear in red on the whiteboard so the Operations Team can see them more easily.

6. Critical Evaluation

The aim of the project was to design an Intranet Operations Board for IT Operations, Sun Microsystems, Germany. The idea for the Operations Board was produced during the developers experience on the placement year with IT Operations. The Operations Board needed to replace an existing whiteboard information system. The new system was developed using the existing physical whiteboard system as a basis.

Using a database to store the Operations Board data was a logical decision, as was the use of a web browser to access this data. The hardware to serve the web pages was already available as the department already has a lightly loaded web server. Aspects of the project which were not immediately obvious were: which package to use for the web server; what database system to use.

Zope was chosen as the web development technology mainly because of its 'through-the-web' editing system and its ability to connect to an external database source. The MySQL database system was chosen mainly due to the fact that it is open source and that it was supported by Zope. In the light of experience both Zope and MySQL proved to be good choices.

User Centred Design involved the end-users in the development of the Operations Board, producing constant feedback and providing in-situ testing of new components. This feedback was invaluable as the Operations Board could then be developed to meet the users' needs. Without this repetitive testing and feedback, as with other design methodologies, the Operations Board would not have been so effective.

The look and feel of the Operations Board is one of the main factors for the effectiveness of the system. Throughout the use of the Operations Board, the

user is presented with the same icons, colours, input boxes and selection lists. This was done consciously to make the Operations Board more usable. The transfer from the existing whiteboard system to the Operations Board was simple and quick. One of the early versions of the Operations Board was adopted and the existing whiteboard was removed. Recently, a large digital projection screen has been introduced to the main office, in Munich, which is dedicated to displaying the whiteboard component of the Operations Board.

There are many advantages of the Operations Board over the existing whiteboard system. The Operations Board can be accessed worldwide within the Sun network, not just in the countries that use it. The Operations Board is used as a task tracking tool. Information about the current progress of a task can be viewed and edited online. This could not be displayed using the existing whiteboard system, due to lack of space. The only disadvantage of the Operations Board over the previous whiteboard system is that it takes longer to create a task. With the whiteboard system a member of the Operations Board simply picked up a pen and wrote it on the board. It requires a few more steps to achieve the same result with the Operations Board.

The main weakness of the Operations Board is the database design. If the Operations Board was to be redesigned, or even redeveloped for a future release, the database structure and its relationships would need to be redesigned. Creating relationships between the entities in the database would obviate at least four data fields per task. This reduction would make the database smaller and more efficient.

A professional web site designer was shown the DTML code within the Operations Board. The designer was impressed by the quality of DTML used. However, many improvements could be introduced. These improvements included using one new portion of DTML code to do the functions of two of the existing functions. This would not increase the speed of the Operations

Board, but make the code more readable for humans. Another improvement was to correctly comment and indent the DTML code, which also would make it more readable by humans.

The implementation of the Operations Board was an all-round success. The tool has been adopted as the department's **only** task tracking tool, and is used in over 15 countries throughout Europe and Africa. The overall level of feedback from the Operations Team and all users across the regions has been very good. The Operations Board is seen as a useful tool by users. An endorsement from Mervyn Sloan (Technical Manager for the region) and Stefan Holtz (Operations Manager for the region) can be found in Appendix A5.

7. References

Project Insight (2003), *Project Insight*, [online]; last accessed on 11th February 2003 at URL: <http://www.projectinsight.com>

Zope, *What is Zope?*, [online], last accessed 16th February 2003 at URL: <http://www.Zope.org/WhatIsZope>

The Apache Foundation, *Apache HTTP Server Project*, [online], last accessed 16th February 2003 at URL: <http://httpd.apache.org/>

Techtarget Network, *WhatIs?com*, [online], last accessed 17th February 2003 at URL: <http://whatis.techtarget.com/>

Techtarget Network, *searchDatabase.com*, [online], last accessed 1st March 2003 at URL: <http://searchdatabase.techtarget.com/>

Netcraft, *Web Server Survey*, [online], last accessed 17th February 2003 at URL: <http://www.netcraft.com/survey/>

Oracle, *Oracle 9i Database Server*, [online], last accessed 20th February 2003 at URL: <http://www.oracle.com/ip/deploy/database/oracle9i/index.html>

MySQL, *MySQL Database Server*, [online], last accessed 20th February 2003 at URL: <http://www.mysql.com/products/mysql/index.html>

PHP, *PHP: Hypertext Pre-Processor*, [online], last accessed 23rd February 2003 at URL: <http://www.php.net/>

Wiley, John & Sons, *Encyclopaedia of Software Engineering*, [online], last accessed 25th February 2003 at URL: http://www.mrw.interscience.wiley.com/ese/ese_articles_fs.html

Boehm, Barry W., (1988), *A Spiral Model of Software Development and Enhancement*, IEEE Computer Society.

Sommerville, Ian, (2001), *Software Engineering*, 6th Ed., Addison-Wesley.

Perl, *A Short Guide to DBI*, [online], last accessed 1st March 2003 at URL: <http://www.perl.com/pub/a/1999/10/DBI.html>

Gadfly, *Frequently Asked Questions*, [online], last accessed 1st March 2003 at URL: <http://gadfly.sourceforge.net/faq.html>

University of Southampton, *HCI Design*, [online], last accessed 1st March 2003 at URL: <http://www.ecs.soton.ac.uk/~nmg/hci/design/>

University of Southampton, *Handbook of User Centred Design*, [online], last accessed 1st March 2003 at URL: <http://www.ejeisa.com/nectar/inuse/6.2/>

Zelle, J. M., & Mooney, R. J., (1994), *Combining top-down and bottom-up methods in inductive logic programming*, New Brunswick.

Sun Microsystems, Sun Marketing Resource Centre, [online], last accessed 28th March 2003 at URL: <http://www.sun.com/smrc>

8. Bibliography

Boehm, Barry W., (1988), *A Spiral Model of Software Development and Enhancement*, IEEE Computer Society.

Sommerville, Ian, (2001), *Software Engineering*, 6th Ed., Addison-Wesley.

9. Appendix

- A1. Project Specification
- A2. Sample code from an SQL method
- A3. Same code from a DTML web page
- A4. Minutes of a telephone conference with Ben Mason (Operations Board developer), Stefan Holtz (Operations Manager) and Paul Blakeman (Web Operations).
- A5. Endorsement from Mervyn Sloan (Technical Manager) and Stefan Holtz (Operations Manager).

A1. Project Specification



Sheffield Hallam University

SCHOOL OF COMPUTING AND MANAGEMENT SCIENCES

BSc (Hons) Computing (Networks and Communications) Final Year

PROJECT DEFINITION

Student: Ben Mason Date: 04/11/2002
Supervisor: Chris Bates
Level of Project: BSc (Hons) Computing (Networks and Communications)
Title of Project: Design and Implementation of an Intranet Operations Board

ELABORATION

World Wide Client Services (WWCS) is a department within Sun Microsystems GmbH, based in Munich, Germany, providing IT support to end users within Sun and maintaining a large network and computer infrastructure. The day to day running of the department, of approximately 20 personnel, includes dealing with a variety of tasks. At present, the department uses a whiteboard to delegate and record current and future tasks. With this in mind, the student will design and implement a web-based database to deal with the scheduling and logging of these tasks in real time. The aim of the project is to completely computerise the information regarding daily tasks and maintenance.

The department employs a duty manager who, amongst other things, manages the existing helpdesk ticketing system. This system provides an interface between the end users and the operations staff. The new 'Operations Board' will also be managed by the duty manager, as a means of distributing the helpdesk tasks and general maintenance.

OBJECTIVES AND DELIVERABLES

The student is required to:

1. Analyse the requirements and existing environment of the department.
2. Identify and discuss web development and database methodologies.
3. Design the system.
4. Implement the database and test functionality.
5. Implement and test web interface.
6. Implement the 'whiteboard' interface.
7. Develop and implement a test strategy.
8. Analyse and implement features and/or problems found during testing.
9. Critically evaluate the project.
10. Suggest future enhancements and developments.
11. Produce a project report.

Deliverable for this project will be a web-based intranet database to organise and record tasks within an IT support department.

TASK PLAN

TASK	DURATION
1. Analyse the requirements and existing environment of the department.	3 days
2. Identify and discuss web development and database methodologies.	3 days
3. Design the system.	4 days
4. Implement the database and test functionality.	5 days
5. Implement and test web interface.	8 days

6. Implement the 'whiteboard' interface.	3 days
7. Develop and implement a test strategy.	3 days
8. Analyse and implement features and/or problems found during testing.	3 days
9. Critical evaluation of the project development and deliverable.	1½ days
10. Write up the project.	4 days
TOTAL	37½ days

MARKS BREAKDOWN

Process

1. Investigation	8%
2. Design	8%
3. Development	24%
TOTAL	40%

Deliverable

1. Quality of description	15%
2. Quality of deliverable	15%
TOTAL	30%

Reflections/critical evaluation	20%
Project report presentation	10%
TOTAL	30%

OVERALL TOTAL 100%

A2. Sample code from an SQL method

The following SQL is used by the whiteboard to retrieve all *tasks* that do not have the status 'Closed', 'Cancelled' or 'Proposal'. They are ordered by *start date* and then *priority*.

```
select tid, country, start_date, hostname, gcn_number,  
priority, owner, approx_end_date, summary from tasks  
where  
tasks.status != 'Closed'  
and  
tasks.status != 'Cancelled'  
and  
tasks.ops_status = 'task'  
and  
tasks.status != 'Proposal'  
order by start_date, priority
```

A3. Sample code from a DTML web page

The following code is used to create the table of tasks when a user clicks on *My Tasks*. If the user does not have any tasks, the following message is displayed **instead** of the table: *There are currently no tasks owned by you.*

The `<dtml-in>` tag retrieves data using SQL methods. In this example, the SQL method used is `SQL_show_my_tasks`.

```
<table border="0" cellpadding="0" cellspacing="0" width="100%">
<tr height=19>
<td bgcolor="#000000"><div class="headerpadding">
&nbsp;<font
color="white"><b>My tasks</b></font>
</div></td>
</tr>
</table>
<p>
<dtml-in SQL_show_my_tasks size=10 start=query_start>

  <dtml-if sequence-start>

<table border="0" cellpadding="0" cellspacing="0"
class="dkgrey1" width="100%">
<form action="<dtml-var URL1>/multi_close" method="get">
<tr><td>
<table border="0" cellpadding="0" cellspacing="1" width="100%"
summary="placeholder summary" valign="middle">
<tr>
<th id="t3c1" class="columncaption"><div
class="headerpadding"><center>&nbsp;</div></th>
<th id="t3c1" class="columncaption"><div
class="headerpadding"><center>&nbsp;</div></th>
<th id="t3c1" class="columncaption"><div
class="headerpadding"><center>Site</div></th>
<th id="t3c2" class="columncaption"><div
class="headerpadding"><center>Start date</div></th>
<th id="t3c3" class="columncaption"><div
class="headerpadding"><center>Priority</div></th>
<th id="t3c3" class="columncaption"><div
class="headerpadding">Summary</div></th>
<th id="t3c3" class="columncaption"><div
class="headerpadding"><center>Tools</div></th>
</tr>

  </dtml-if sequence-start>

<tr <dtml-if sequence-odd>class="yellow1" <dtml-
else>class="yellow2"</dtml-if>>

<td headers="t3c1" width="6%"><div class="pad5x10">
```

```
<center>
<input type="checkbox" name="tid" value="<dtml-var tid>">
</center>
</div></td>

<td headers="t3c1" width="6%"><div class="pad5x10">
<center>
<a href="<dtml-var URL1>/show_country_tasks?hcountry=<dtml-var
country>"></a>
</center>
</div></td>

<td headers="t3c1" width="10%"><div class="pad5x10">
<center>

<dtml-var site>

</center>
</div></td>
<td headers="t3c2" width="15%"><div class="pad5x10">
<center>
<dtml-var start_date>
</center>
</div></td>

<td headers="t3c3" width="6%"><div class="pad5x10">
<dtml-if "priority=='P1'">
<font color="red"><b>
</dtml-if>
<dtml-if "priority=='P3' or priority=='P4' or priority=='P2'">
<b>
</dtml-if>
<center><dtml-var priority></center>
</div></td>

<td headers="t3c3" width="42%"><div class="pad5x10">
<dtml-if summary>

<dtml-var summary size=50>

<dtml-else>
none
</dtml-if>
</div></td>

<dtml-var tools_html>

</tr>
  <dtml-if sequence-end>
</table>
</td></tr>
</table>

<dtml-var next_prev_links>
</dtml-if sequence-end>

<dtml-else>
  There are currently no tasks owned by you.
</dtml-in>
```

A4. Minutes of a telephone conference.

The following minutes were taken during a telephone conference between Ben Mason (Operations Board developer), Stefan Holtz (Operations Manager) and Paul Blakeman (Web Operations). The meeting was held to discuss new features and to prioritise the implementation of these features.

Dear All

Following a conf. call today between Ben Mason (Ops Board author), Paul & Stefan, the following outcome was reached regarding the new Operations Tool.

The Agenda

=====

1. Stefan outlined what he felt was missing, in particular - the "Whiteboard".
2. Paul relayed his & other suggestions from team members.
3. All three decided on priorities of jobs to be done.

The Outcome

=====

WHITEBOARD - priority=HIGH

- 1) Each 'job' should display the following:

 /Country/Owner/Hostname/GCN or Radiance or ServiceDesk
NUMBER/Date/Priority/Summary/

- 2) When you 'click' the flag of a job this shall display task details in new window.

- 3) 'Jobs' should be displayed in the following categories (just like current ops board)

 Planned Outages
 Unplanned Outages
 Maintenance
 Completed Tasks (last 24hrs)

Due to limitations of space (ie: monitor size) P1's and P2's are the only 'tasks' to be shown.

Try to avoid a 'scroll-bar' appearing in this instance.

4) Any task/job that is 'late' should be labelled or displayed as ISSUE.

The following 3 areas will realise full potential in the Ops Team meetings. Although individuals who are logged in can change information about their own tasks please remember we need to quickly assign and complete/close tasks.

This last point is more relevant to UNASSIGNED TASKS & PROPOSED TASKS.

ADD TASKS (greater functionality) - priority=HIGH

+++++

1) When selecting 'OPS type' need the following selections available.

- Maintenance
- Planned Outage
- Unplanned Outage

2) When selecting 'Status' we need the following option added.

- Proposal

3) Need to be able to set a 'task' that is NOT host/machine based.

UNASSIGNED TASKS (greater functionality) - priority=HIGH

+++++

1) Need to be able to view this in Ops Team meeting and assign a task to anybody

. At present there is only 'Assign To Me' option.

PROPOSED TASKS (new functionality) - priority=HIGH

+++++

1) Need a viewing option, exactly like above, but where we can assign a task to team member.

2) Need to be able to assign (or even close) proposals.

As mentioned we will have the tool so anybody can change everybody

elses.

The above will be implemented by close of business tomorrow. (Dec 12)

As of Friday (Dec 13) all tasks proposals or current tasks must be entered into the new tool.

At the Ops meeting on Monday WE WILL USE THIS TOOL.

Any functionality or data in the ProjectBoard Tool concerning 'tasks', will be transfered on Friday. (Dec 13)

Below are points which will be implemented from the start of next week (wk 51) and completed by Jan 6th.

priority=LOW

- 1) Server data (from RegTool) will be gathered automatically and uploaded every night.
 - 2) Multi-Task has slight bug in that you have to close each server job individually!
e.g. Select 100 servers in one go - 'add' each job individually.
 - 3) Edit hosts page should display all on one page.
Add a search feature to 'jump' to a particular one.
 - 4) Add a link to GCN number (exists already for Radiance and service desk).
 - 5) Delete the 'server information' link in admin section.
-

A5. Endorsement from Mervyn Sloan and Stefan Holtz

The following page shows a letter from Mervyn Sloan (Technical Manager for the region) and Stefan Holtz (Operations Manager) discussing the Operations Board and its development.